



11/28/00

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.

36.P292

First Named Inventor or Application Identifier

MOIS-612, 614-615

Express Mail Label No.

EL714839434US

JC653 U.S. PTO
09/123976
11/28/00

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO:

Commissioner for Patents
Box Patent Application
Washington, DC 202311. ☒ Fee Transmittal Form
(Submit an original, and a duplicate for fee processing)2. ☐ Applicant claims small entity status.
See 37 CFR 1.27.3. ☒ Specification Total Pages 444. ☒ Drawing(s) (35 USC 113) Total Sheets 95. ☐ Oath or Declaration Total Pages a. ☐ Newly executed (original or copy)b. ☐ Copy from a prior application (37 CFR 1.63(d))
(for continuation/divisional with Box 17 completed)i. ☐ **DELETION OF INVENTOR(S)**
Signed Statement attached deleting inventor(s)
named in the prior application, see 37 CFR
1.63(d)(2) and 1.33(b).6. ☐ Application Data Sheet. See 37 CFR 1.78
"Express Mail" mailing label number EL714839434US
Date of Deposit 11/28/00I hereby certify that this paper or fee is being deposited with
the United States Postal Service "Express Mail Post Office to
Addressee" service under 37 CFR 1.10 on the date indicated above
and is addressed to the Commissioner of Patents and Trademarks,
Washington, D. C. 20231.KRISTY KAPPELMAN
(Typed or printed name of person mailing paper or fee)Kristy Kappelman
(Signature of person mailing paper or fee)7. ☐ CD-ROM or CD-R in duplicate, large table or Computer
Program (Appendix)8. ☐ Nucleotide and/or Amino Acid Sequence Submission
(if applicable, all necessary)a. ☐ Computer Readable Form (CRF)

b. Specification Sequence Listing on:

i. ☐ CD-ROM or CD-R (2 copies); orii. ☐ paperc. ☐ Statements verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

9. ☐ Assignment Papers (cover sheet & document(s))10. ☐ 37 CFR 3.73(b) Statement ☐ Power of Attorney
(when there is an assignee)11. ☐ English Translation Document (if applicable)12. ☒ Information Disclosure 9 Copies of IDS
Statement (IDS/PTO-1449) Citations13. ☐ Preliminary Amendment14. ☐ Return Receipt Postcard (MPEP 503)
(Should be specifically itemized)15. ☐ Certified Copy of Priority Document(s)
(if foreign priority is claimed)16. ☐ Other: _____

17. If a CONTINUING APPLICATION, check appropriate box and supply the requisite information:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.
 Prior application information: Examiner Group/Art Unit:

For CONTINUATION OR DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

18. CORRESPONDENCE ADDRESS

☒ Customer Number or Bar Code Label 05514 (insert Customer No. or Attach bar code label here) or ☐ Correspondence address below

NAME

Address

City

Country

State

Telephone

Zip Code

Fax

CLAIMS	(1) FOR	(2) NUMBER FILED	(3) NUMBER EXTRA	(4) RATE	(5) CALCULATIONS
TOTAL CLAIMS (37 CFR 1.16(c))		62 -20 =	42	X \$ 18.00 =	\$ 756.00
INDEPENDENT CLAIMS (37 CFR 1.16(b))		8 -3 =	5	X \$ 80.00 =	\$ 400.00
MULTIPLE DEPENDENT CLAIMS (if applicable) (37 CFR 1.16(d))				\$270.00 =	None
				BASIC FEE (37 CFR 1.16(a))	\$710.00
Total of above Calculations =					\$1,866.00
Reduction by 50% for filing by small entity (Note 37 CFR 1.9, 1.27, 1.28).					
TOTAL =					\$1,866.00

19. Small entity status

- a. ☐ A small entity statement is enclosed
- b. ☐ A small entity statement was filed in the prior nonprovisional application and such status is still proper and desired.
- c. ☐ Is no longer claimed.

20. ☒ A check in the amount of \$ 1,866.00 to cover the filing fee is enclosed.21. ☐ A check in the amount of \$ _____ to cover the recordal fee is enclosed.

22. The Commissioner is hereby authorized to credit overpayments or charge the following fees to Deposit Account No. 06-1205:

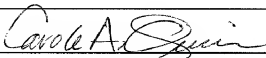
- a. ☒ Fees required under 37 CFR 1.16.
- b. ☒ Fees required under 37 CFR 1.17.
- c. ☐ Fees required under 37 CFR 1.18.

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

NAME

Carole A. Quinn, Reg. No. 39,000

SIGNATURE



DATE

November 28, 2000

"Express Mail" mailing label number EL714839434 US

Date of Deposit: 11/28/00

I hereby certify that this paper or tape is being submitted to the United States Patent Service "Express Mail, First Class to Addressee" service under 27 C.F.R. § 1.10 and that it is being submitted and is addressed to the Commissioner of Patents and Trademarks, Washington, D. C. 20531.

KRISTY KAPPELMAN

(Typed or printed name of person making deposit or file)

Kristy Kappelman

(Signature of person making deposit or file)

DIGITAL IMAGE RETRIEVAL AND STORAGE BACKGROUND OF THE INVENTION

Field Of The Invention

5

10

15

The present invention relates to the retrieval and storage of image data captured using a digital image acquisition device such as a still camera, a video camera or a scanning device, and of image data stored on removable storage media such as compact flash, smart and memory stick media. An image retrieval architecture is described such that image data may be retrieved from any digital image acquisition device and/or from removable storage media, and a "one-click" image upload operation is available to minimize an amount of interaction needed to retrieve image data from the digital image acquisition device.

Description Of The Related Art

Digital cameras are becoming more popular, particularly as the quality of a captured image begins to approach the quality of analog cameras (e.g., 35MM cameras). However, there are still some drawbacks with digital cameras. For example, a digital camera has limited storage capacity which gives rise to a need to be able to transfer images captured by the digital camera to external storage. In addition, unlike analog images, the procedures for processing images require the user to use some sort of computing equipment (e.g., personal computer, or PC) to retrieve an image in the camera's storage and/or process a retrieved image.

Typically, a digital camera user must individually retrieve each image saved in the camera's storage for storage on the PC. The PC may be owned by the user or an acquaintance, or the user may make use of one of the photo kiosks located in a publicly-accessible place (e.g., stores, hotels, etc.). Individual selection of each image that is to be uploaded from the camera can be tedious and time consuming, and may be especially undesirable when the images are being uploaded using someone else's PC or a photo kiosk.

Further, a user may wish to upload the image to a server (e.g., an Internet server) such as a photo web server to store and/or process the image, to generate photo quality prints, or to share the image with other site users. To make use of the photo web server's capabilities, a user must first

upload an image to a PC, then log onto the Internet, locate the server, locate the image stored on the PC that is to be uploaded and select an upload operation. The user uses client software (e.g., a browser) that executes on the PC to provide an interface with the photo web server that steps the user through the upload process. Once the user selects the submit button in the interface, the browser initiates a push operation to push the image onto the server. The photo web server receives the uploaded image and stores it on a server.

However, the conventional mechanisms used to upload a digital image have drawbacks. For example, in order to upload an image, the user must own or have access to a PC. In addition, the upload process involves multiple steps to first upload to the PC and then to a server. If the server is too busy to receive the image at the time it is initiated by the user, the user must repeat some or all of the previously-described upload steps. Further, the conventional mechanism assumes that the client PC has sufficient resources to store an image and to execute browser software that is capable of pushing an image from the PC's storage to the server.

U.S. Patent No. 5,606,365, issued to Maurinus et al., describes a technique for transmitting sets of raw image information along with a camera identification code, or ID, from a Home Interface Control (HIC) such as a Set Top Box (STB) through an interactive television network to

an interactive control node. The camera's ID is used to retrieve corrections and processing software for the camera, and the corrections are made to the image data using the camera's correction information and processing software. The correction information and processing software are either provided by the camera's manufacturer or the camera user.

Maurinus indicates that computing equipment that many households already have in the home, an HIC, may be used to upload image data. However, Maurinus identifies many uncertainties with respect to the manner in which the images are to be uploaded, using an HIC, to a server or cable head end (CHE). In fact, according to Maurinus, the configuration of an HIC had not yet been determined and Maurinus could not provide such specifics. In fact, Maurinus speculates that an HIC will likely be based on a powerful microprocessor-based computer. However, in actuality, an HIC, such as that provided by Scientific-American, Inc. and General Instrument Corporation, typically has limited processing resources.

In addition, according to Maurinus, to use the system a user must first register a camera and, as a result, an identification code, or ID, that identifies the camera is associated with the camera's correction information and software for correcting an image using the correction information.

Maurinus is silent with regard to an actual mechanism for uploading images from the

camera to the HIC and to the control node. Maurinus is silent with regard to uploading images using an HIC where the HIC may need to be configured for use with multiple, different cameras.

5 Thus, it would be beneficial to have an image retrieval capability that provides an approach for uploading images, and that may be used with any manufacturer's cameras regardless of whether or not a camera has first been registered with a service
10 provider.

SUMMARY OF THE INVENTION

 The present invention relates to the retrieval and storage of image data captured using a
15 digital image acquisition device such as a still camera, a video camera or a scanning device, and of image data stored on removable storage media such as compact flash, smart and memory stick media. An image retrieval architecture is described such that
20 image data may be retrieved from any digital image acquisition device and/or from removable storage media, and a "one-click" operation is available to minimize an amount of interaction needed to retrieve image data from the digital image acquisition
25 device.

 In an aspect of the invention, a facility is provided for uploading images directly from a device to a server using upload software that executes on client computing equipment. According
30 to this aspect of the invention, a broadband network such as that used to transmit broadcast (e.g.,

television) signals to a home. The present invention accommodates an upload bandwidth that may be just a fraction of the downstream, and provides a network centric storage and image manipulation capability.

The client upload software comprises two components, one which remains resident on the client, and another which is loaded when the client detects that a physical connection has been established with a device. The dynamically-loadable client upload software may be retrieved locally or from a server, and is selected based on the identification information of a device that is connected to the client.

The client upload software is configured to interface with driver software executing in the client (PC, STB, HIC, etc.), and interfaces with a server (e.g., a CHE, photo web server, etc.) during an upload operation. The driver software communicates with the device to retrieve digital images stored in the device.

Initially, the client upload software operates to establish a connection (e.g., a streaming connection) with the server. An upload proxy component executing on the server then manages the upload process so that the digital image data is retrieved from the device by the client and transmitted to the server. In effect, the client may act as a temporary buffer between the device and the server during the upload process. The server may postpone an upload if, for example, the server

or network is otherwise busy. Once on the server, a digital image is stored in a data store by the upload proxy and may be retrieved for processing (i.e., image editing, printing, etc.).

5 Advantageously, the upload facility of the present invention allows a user to upload one or more images to a server. The operation appears to the user as a single operation, or step. There is no need for the user to first make a persistent, intermediate copy of each image at the client
10 computing equipment and then initiate an upload of the stored image from the client to the server. Since a temporary buffer, or other storage, is all that is needed for the upload to a server, the
15 client computing equipment needs minimal storage capacity.

 Of course, it is possible with the present invention, to allocate memory according to its availability such that the client is able to store a
20 persistent copy which may then be transmitted to the server, or processing in some manner. In so doing, it would be possible to allow the user to disconnect the device when transmission is delayed and before the images are actually uploaded to the server.
25 However, in a case that the client computer has limited storage capacity, it is optionally provided to allow the device to remain connected to the client computer.

 Regardless of whether the device remains
30 connected, an image, or images, may be uploaded at a time when the server becomes available. That is,

the server may initiate an operation to "pull" the image data at a time determined by the server. In a case that the device remains connected to the client computer, the image data may be retrieved from the device in response to an initiation of a pull operation by the server. Where the data is stored by the client, the persistent copy of the image data is used to transferred to the server.

To further accommodate a client computer's resource constraints or capabilities, the client upload software is preferably split into two components. A resident component and a dynamically-loadable component which is likely to be the larger of the two components. The dynamically-loadable component is initiated when an upload operation is to be performed, and need not be resident when no operation is being performed thereby freeing memory for other uses. In addition, to accommodate different digital image acquisition devices, different dynamically-loadable components may be written, and an appropriate one selected for uploading an image from the device. While a universal dynamically-loadable component may be used with the present invention, components geared for use with one or more specific devices may reduce the size of each dynamically-loadable component and consequently the amount of the client's memory that is needed for the component. It is possible with the present invention to have dynamically-loadable components for each device or a group of devices (e.g., devices from the same manufacturer).

Another aspect of the present invention is directed to a technique of uploading images from a device using a single selection, or click. A user interface is provided that includes a "One Click Upload" option that is selectable by, for example, a click of a button (e.g., of a mouse, television remote control or other pointing device) when the cursor is positioned on the option. In response to the single click, all of the images that are currently stored on the device are retrieved from the device and stored in an external image database. A unique identifier, or token, is generated which is associated with the stored images, and the token is saved in the device's storage. Thereafter, the images may be retrieved from the external storage using the token stored in the device. That is, the token is retrieved from the device and used to retrieve the associated images from the image database. The images may then displayed in such a way that the user may scroll through them to identify the ones that are to be retained or deleted.

Thus, by using the present invention, the user interaction needed to initially retrieve the images from a digital image acquisition device is minimized, and the browsing and selection of images may be deferred to a more convenient time and/or setting.

This brief summary has been provided so that the nature of the invention may be understood quickly. A more complete understanding of the

invention can be obtained by reference to the following detailed description of the preferred embodiment(s) thereof in connection with the attached drawings.

5 BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a representation of a cable broadband network embodying the present invention.

Figure 2 illustrates representative software architecture of a set top box.

10 Figure 3 provides an overview of an architecture for use in transferring digital image data and associated information between a cable head end and a set to box according to the present invention.

15 Figures 4A to 4C illustrate a flow diagram of process steps to upload images to a cable head end using a dynamically loadable module executing on a set top box according to the present invention.

20 Figure 5 provides a more detailed example of locating loadable device module 304 according to the present invention.

 Figures 6A through 6D provide an example of a user interface according to the present invention.

25 Figures 7A and 7B provide an overview of a "one-click" operation according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

30 Figure 1 illustrates a regional broadband digital cable network connected to the internet and utilizing the present invention. The network is

capable of delivering analog and digital broadcasts, secure analog and digital broadcasts, analog and digital pay-per-view, analog and digital impulse pay-per-view, digital near video on demand, one-way
5 real-time datagram (broadcast IP data packets), and two-way real-time datagram (addressed IP data packets). In addition, according to the present invention, the regional broadband digital cable network is used to transmit digital image data and
10 associated information.

As shown in Figure 1, the above-listed broadcast services may be delivered by value-added service provider systems and network control systems (not shown) located at cable head end (CHE) 6.
15 Value-added service provider systems include digital satellite distribution systems, applications executing on cable servers (such as special-purpose applications like subscriber service application, content gather applications, etc.) and digital media
20 servers outputting MPEG-2 datastreams. Network control systems provide management and control for the services supported by the broadband network.

Alternatively, services may be delivered from World Wide Web (WWW) 4 through internet proxy
25 5, for example, from internet site 8. Examples of internet site 8 include photo service providers, banking, retailing, utilities, and the like.

In either case, the services are delivered to Cable Head End (CHE) 6, which serves as an
30 interface between the service providers and the rest of the broadband network.

In particular, CHE 6, which is responsible for providing services to multiple homes (e.g., approximately 500,000 to 1,000,000), is connected via fiber optic cabling to hubs 7, which are
5 connected to CHE 6 or other hubs 7. Each hub 7 is, in turn, connected to at least one node 9, also using fiber optic cabling. Coaxial cable is then used to connect each node to Set Top Boxes (STB's) 10. For instance, the STB 10 may comprise one of
10 several currently available STB models available such as Scientific Atlanta's Explorer series and General Instrument's DCT-2000 and DCT-5000+ models. While the present invention is described with regard to a STB, it should be apparent that any type of
15 home interface control (HIC) that interfaces with a broadband network such as a digital cable network may be used. Finally, each STB 10 is connected to television 11 as well as other devices not shown (e.g., printer). Accordingly, services are
20 delivered from a service provider to CHE 6, to one or more hubs 7, to node 9, to STB 10 and to television 11.

It should be noted that, by virtue of the foregoing arrangement, a service infrastructure may
25 be distributed among CHE 6, hubs 7, or other facilities. Further, while the present invention is described with reference to a digital cable network of Figure 1, it should be apparent that any
broadband network interconnection may also be used
30 between a client such as STB 10 and a server such as CHE 6. For example, it is possible to use a

satellite, or other connection, with the present invention.

Figure 2 illustrates representative software architecture of set top box 10. In general, this software architecture, together with the hardware architecture of the set top box, supports, in addition to the reception of analog and digital services, the transmission of digital image data and associated information.

Through the software architecture illustrated in Figure 2, STB 10 hosts various applications that present to the home user functionality offered by various cable services. Typical applications are a navigator, an interactive program guide, electronic mail and a web browser. Most of these applications are client/server implementations, where STB 10 hosts the client software, and CHE 6 hosts the server software. Communication between client and server over the cable network is facilitated by an operating system executed on STB 10, and is performed through API's. One example of an API that may be used to communicate between CHE 6 and STB 10 is a broadband protocol transport (BPT) which is offered by Canon Information Systems, Inc. of Irvine, California. However, it should be apparent that any protocol suitable for use over a broadband network may be used with the present invention As is described herein, CHE 6 and STB 10 may communicate using the Hypertext Transfer Protocol (HTTP). Depending on the hardware platform and the operating system,

applications may be resident at STB 10, or can be downloaded from a remote site including servers situated at, or available via, CHE 6 for execution at STB 10.

5 As shown in Figure 2, software architecture and STB 10 includes an interface 31 to hardware, an operating system 32, an HTML client 34, resident applications 35, and other applications 36. As is described in more detail below, resident
10 applications 35 includes a resident device module that operates to cause a loadable device module to be loaded. The loadable device module is configured to communicate with CHE 6 to transfer digital image data and associated information between STB 10 and
15 CHE 6.

 The operating system 32 is usually vendor-specific for the STB, and may include operating systems such as PowerTV, WinCE, MicroWare or OpenTV. HTML client 34 provides a group of independent
20 handlers that can be plugged together in conformity to known plug-in specifications so as to provide ability to handle different types of media such as HTML, GIF, MPEG, HTTP, Java script, etc. The HTML client 34 is used to allow STB 10 to render HTML
25 documents to a windows manager for display on the local television receiver. HTML documents may be retrieved from local cache, from in-band and out-of-band broadcast carrousels, VBI streams, HTTP proxy servers located at CHE 6, or remote HTTP servers
30 accessed by STB 10 over the internet. In the latter case, documents retrieved from external web servers

are filtered by a proxy according to predefined filtering criteria (such as surf watch), which also may convert requested documents into formats supported by the HTML client 34.

5 Resident applications 35 include such applications as the aforementioned resident device module, navigator, interactive program guide, and the like. Applications 35 and 36 include a web browser, an e-mail program, loadable device module, 10 and the like.

 Figure 3 provides an overview of an architecture for use in transferring digital image data and associated information between a cable head end and a set top box according to the present 15 invention.

 As part of hardware interface 31, device driver 302 provides an interface to device 301 external to STB 10 and connected via an interface to STB 10. Examples of such an interface include a 20 Universal Serial Bus (USB), parallel and a IEEE 1394 interface connection. Device 301 is, for example, a digital device such as a still or video camera or a scanning device. Alternatively, device 301 may be a device (e.g., a reader) that is capable of 25 retrieving data stored on a removable storage media such as a compact flash card, smart media or memory stick media, for example. Thus, device 301 is a digital image storage device that may or may not additionally be able to acquire, or capture, digital 30 image data. Device driver 302 is configured to send and receive messages to and from device 301.

Device driver 302 detects device 301 when it is plugged into an external interface of STB 10. Information received by device driver 302 from device 301 is forwarded to resident device module 5 303. Such information includes type, or identification, information such as manufacturer and product identification information.

As is described in more detail below, resident device module 303 identifies an appropriate 10 loadable device module 304 for use in communicating with device 301 and CHE 6 to transfer image data and associated information. Where the identified loadable device module 304 is not available on STB 10, resident device module 303 causes loadable 15 device module 304 to be transferred to STB 10 and initiated on STB 10.

Loadable device module 304 communicates with upload server via a broadband protocol transport (BPT) application programming interface 20 (API) 305 that interfaces with broadband transport client 306 and broadband transport server 308 components executing on STB 10 and CHE 6, respectively. Communication channel 307 is formed over broadband digital cable network and is used to 25 transmit such information as device settings, images and queries, for example. In addition to transmitting image data and associated information, communication channel 307 may be used to request and receive loadable device module 304.

30 Image data received by upload server 310 is stored in database (or data store) 312. Requests

for stored image data may be received by imaging server 313. For example, HTTP client 316 executing on STB 10 may forward a request (e.g., an Hypertext Markup Language, or HTML, request), via communication channel 315, for one or more images via HTTP server 314 to imaging server 313. In addition, imaging server 313 may receive an image request from an such internet sites as photo print service provider 318. Imaging server 313 retrieves images stored in database 312 in response to such request.

Figures 4A to 4C illustrate a flow diagram of process steps to upload images to a cable head end using a dynamically loadable module executing on a set top box according to the present invention.

Steps S402 through S405 are performed by resident device module 303 to identify loadable device module 304 and cause it to be initiated in STB 10. Loadable device module 304 performs steps S406 through S429 to transfer image data and associated information between STB 10 and CHE 6.

Referring to Figure 4A, a determination is made whether a device has been detected. Resident device module 303 receives a message from device driver 302 when device 301 is connected which includes device information received from device 301. Resident device module 303 uses the device information to identify the type of device at step S403. For example, device 301 can transmit a manufacturer and model as device information. The

device information is forwarded to resident device module 303 by device driver 302.

At step S404, resident device module 303 determines whether an appropriate loadable device module 304 is already resident in memory of STB 10. If not, processing continues at step S405 to locate loadable device module 304 in local storage of STB 10, or request loadable device module 304 from another location (e.g., CHE 6 or a server connected to CHE 6 via the internet).

Figure 5 provides a more detailed example of locating loadable device module 304 according to the present invention. At power up, device driver 302 and 303 are initiated and a list of device's that are supported are sent by resident device module 303 to device driver 302.

When device 301 is detected by device driver 302, device driver 302 uses the information received from device 301 to identify a device owner. Device driver 302 forwards the device information including device manufacturer and model information to resident device module 303. Resident device module 303 identifies an appropriate loadable device module 304 for use with the connected device 301 using, for example, a mapping mechanism (e.g., a table) that associates a manufacture, model or both with an instance of loadable device module 304.

Resident device module 303 determines whether the identified loadable device module 304 is already executing in STB 10. For example, loadable device module 304 may be loaded at power up, or may

have been loaded to service this, or another, connected device 301. Loadable device module 304 may be loaded based on a user preference, for example.

5 If loadable device module 304 is not executing in STB 10 but is available from local storage, resident device module 303 causes loadable device module 304 from local storage. Otherwise, resident device module 303 connects to the
10 identified location and requests resident device module 303. For example, resident device module 303 connects to CHE 6 via BPT API 305 and broadband transport client 305 to send a request for loadable device module 304. Upload server 310 can respond by
15 sending loadable device module 304 to STB 10. Where loadable device module 304 is not available from storage of CHE 6, CHE 6 may request loadable device module 304 from internet site 8.

 CHE 6 forwards loadable device module 304
20 to STB 10 and resident device module 303 which causes loadable device module 304 to begin execution in STB 10.

 Referring again to Figure 4A, loadable device module 304 identifies at step S406 an amount
25 of available memory. At step S407, a determination is made whether there is sufficient available memory. Loadable device module 304, may elect to allocate enough memory to temporarily store all of the images that are to be uploaded to CHE 6 in a
30 case, for example, CHE 6 is unavailable for an upload operation due to network congestion or CHE 6

is otherwise busy. Alternatively, loadable device module 304 may elect to allocate memory for storing some portion of the data so that data may be momentarily buffered when it is retrieved from device 301 before it is uploaded to CHE 6. The latter option may be selected where there is a limited amount of memory.

If it is determined by loadable device module 304, at step S407, that there is insufficient available memory, an error condition is raised and processing continues at step S428 to deallocate memory and terminate loadable device module 304.

It is determined, at step S407, that there is sufficient available memory, processing continues at step S410 to allocate an amount of memory determined by loadable device module 304. Processing then continues at step S412 to determine whether a connection has been established with CHE 6. In a case such as a broadband digital cable network, a connection established between CHE 6 and STB 10 may remain open between transfers. If a connection has already been established, processing continues at step S414. If not, processing continues at step S413 to establish a connection between STB 10 and CHE 6.

At step 414, loadable device module 304 inquires into the availability of CHE 6 to receive data transmissions. If it is determined at step S415 that CHE 6 is available, processing continues at step S416 to retrieve the data (e.g., image data) from device 303 and upload it to CHE 6. More

particularly, loadable device module 304 retrieves image data and forwards the data along with a request to generate a broadband transport message to broadband transport client 306 via BPT API 305.

5 Broadband transport client 306 transmits a formed message to broadband transport server 308 via BPT API 305. The data contained in a received broadband transport message is forwarded by broadband transport server 308 to upload server 310.

10 In addition to transmitting image data, Broadband transport server 308 and broadband transport client 306 may communicate with acknowledgments indicating that messages are received without error.

15 Once all of the image data has been retrieved from device 301 and transferred to CHE 6, processing continues at step S425 to await a reply from CHE 6 that all of the data has been received successfully. In addition, as is described in more
20 detail below, CHE 6 may return information associated with the uploaded data (e.g., a identification and/or storage location) that is to be forwarded to device 301. Processing continues at step S426 to determine whether CHE 6 returned such
25 information. If so, processing continues at step S427 to send the data received from CHE 6 to device 301. Processing then continues at steps S428 and S429 to deallocate memory and terminate loadable device module 304.

30 If it is determined, at step S415, that the server is not ready to receive the data to be

uploaded, processing continues at step S418 to determine whether the data to be retrieved from device 301 may be stored in memory of STB 10. Loadable device memory 304 may determine whether the amount of memory allocated in step S410 is sufficient to store the image data. If not, loadable device module 304 may attempt to allocate additional memory. In any case, if there is insufficient memory to store the data, processing continues at step S419 to display a status to the user. Processing continues at steps S428 and S429 to deallocate memory and terminate loadable device module 304.

Alternatively, where it is undesirable for STB 10 to store the image data, a user may be given the option to leave device 301 connected to STB 10, and wait for CHE 6 to initiate an upload. In such a case, once CHE 6 initiates the upload, processing continues at step S416 to retrieve the data from device 301 and upload it to CHE 6.

If it is determined at step S418 that there sufficient local storage, loadable device module 304 retrieves the data from device 301 via device driver 302 at step S421 and stores the retrieved data locally. A status (e.g., success or failure) of the data retrieval may be displayed to the user.

In a case that CHE 6 previously indicated a "not ready" status (e.g., in a case that CHE 6 and/or the network is busy), it notifies STB 10 when it is ready to receive the data. At step S422,

loadable device module 304 awaits the availability of CHE 6. When a "ready" indication is received, processing continues at step S423 to upload the stored data to CHE 6. In such a case, the data
5 previously retrieved from device 301 is forwarded to broadband transport client 306 to broadband transport server 308 via BPT API 305. Alternatively, where device 301 remains connected to STB 10, loadable device module 304 may retrieve the
10 data from device 301.

Broadband transport server 308 forwards the uploaded data to upload server 310. Processing continues at step S425 to await a status at CHE 6, forward data received from upload server 310 to
15 device 301 and terminate loadable device module 304 as previously described.

A user interface is provided with the present invention in response to a connection of device 301 being detected. Figures 6A through 6D
20 provide an example of a user interface according to the present invention.

Referring to Figure 6A, a user is presented with the options, in display 601, of logging in, or initiating a "one-click" image upload
25 operation. By selecting the "one-click" operation, images may be uploaded from device 301 using a single selection, or click. The "One Click" option is selectable by, for example, a click of a button (e.g., of a mouse, television remote control or
30 other pointing device) when the cursor is positioned on, or over, option 301.

In response to the single click, all of the images that are currently stored on device 301 are retrieved from the camera and stored in an image database. A unique identifier, or token, is generated which is associated with the stored images, and the token is saved in storage of device 301. Thereafter, the images may be retrieved from storage using the token stored in device 301. That is, the token is retrieved from device 301 and used to retrieve the associated images from storage. The images may then be displayed in such a way that the user may scroll through them to identify the ones that are to be retained or deleted.

For example, referring to Figure 6D, thumbnail images 632 of the uploaded images may be displayed in display 631. The user may select some or all of the images for deletion or for storage in a new or existing album using options 633 through 636.

Display 611 of Figure 6B allows a user to login by entering a user name and password in input fields 612 and 613, respectively. Alternatively, it may only be necessary to specify a user name. In either case, a user name may be specified using a keyboard or a pointing device to identify the user name one character at a time, or from a list of displayed user names. A password may be entered using an input device such as a keyboard or pointing device as well.

If a valid user name (and password, if required) is entered, display 621 of Figure 6C is

displayed which allows a user to get images stored in an album using selection option 623. In addition, a user may retrieve images uploaded to CHE 6 via a "one-click" operation by selection option 622. When option 622 is selected, a token that is stored in connected device 301 is retrieved from device 301 and used to retrieve the uploaded images.

Figures 7A and 7B provide an overview of a "one-click" operation according to the present invention. Referring to Figure 7A, image data and device identification (or settings) are retrieved from device 301 by client 701. Client 701 may be STB 10 which executes loadable device module 304 and HTML client 316, for example. Client 701 transfers the retrieved image data and settings to server 702. Server 702 may be CHE 6 which executes upload proxy 310, imaging server 313 and HTTP server 314, for example.

As discussed above, communication channel 307 may be used to upload image data and setting from client 701 to server 702, and to return a token for storage by device 301. Communication channel 315 may be used to transfer a user interface such as that described above with reference to Figures 6A through 6D. In addition, communication channel 315 may be used to transmit a user's requests for "one-click" uploaded images. Preferably, communication channel 307 is used to transfer the images to client 701. HTTP client 316 receives HTML page definitions from HTTP server 314, and generates a display (e.g., displays 601, 611, 621 and 631).

The HTML page definitions may be provided by HTTP server 314, imaging server 313 or both.

5 Server 702 generates a token which is associated with the uploaded image data using the settings retrieved from device 301. The token may be, for example, an identifier associated with device 301, the user or both. In addition, the token includes an identifier that uniquely identifies the image data uploaded in the current
10 "one-click" operation.

 The uploaded image data along with an associated token are then saved to storage in database, or datastore, 703. The token is transmitted by server 702 to client 701. Client 701
15 causes the token to be stored in storage of device 301. Preferably, the token is stored in place of, or as a substitute for, the uploaded image data to which the token is associated thereby freeing storage of device 301 for other usage (e.g., storage
20 of additional images).

 A user may request the image data uploaded via a "one-click" operation. For example, a user may select option 622 of display 621. In response to such a selection, client 701 retrieves the stored
25 token from device 301. Client 701 forwards the retrieved token to server 702. Server 702 retrieves the images associated with the token from database 703. The images (e.g., thumbnail images) are sent to client 701 which displays the images as in
30 display 631, for example.

Operations including image deletions,
image selections for inclusion in an album and/or
album creations are transmitted to server 702.
Server 702 saves those images selected by the user
5 for inclusion in an album in database 703 with an
association to a selected album.

In this regard, the invention has been
described with respect to particular illustrative
embodiments. However, it is to be understood that
10 the invention is not limited to the above-described
embodiments and that various changes and
modifications may be made by those of ordinary skill
in the art without departing from the spirit and the
scope of the invention.

WHAT IS CLAIMED IS:

1. A method of retrieving
digitally-captured image data from storage, the
method comprising:

5 detecting a signal from a digital image
storage device, the digital image storage device
capable of accessing the digitally-captured image
data;

10 identifying a component configured to
upload digitally-captured image data, the
identification based on type information received
from the digital image storage device;

15 executing the upload component to cause
the digitally-captured image data to be retrieved
from the digital image storage device and
transmitted to an image repository.

20 2. A method according to Claim 1, wherein
identifying a component configured to upload the
digitally-captured image data is initiated in
response to detection of the digital image storage
device and receipt of the type information from the
digital image storage device.

25 3. A method according to Claim 1, wherein
the digital image storage device is a digital
camera.

30 4. A method according to Claim 1, wherein
the digital image storage device is a scanning
device.

5. A method according to Claim 1, wherein the digital image storage device is a video device.

6. A method according to Claim 1, wherein
5 the digital image storage device is a removable storage media reader.

7. A method according to Claim 1, wherein
10 identifying a component for use in uploading digitally-captured image data further comprises:

identifying the component using a mapping between the upload component and the type information received from the digital image storage device;

15 identifying a location of the upload component; and

retrieving the upload component from the identified location.

20 8. A method according to Claim 1, wherein the type identifies a manufacturer of the digital image storage device.

25 9. A method according to Claim 1, wherein the type identifies a model of the digital image storage device.

10. A method according to Claim 1, wherein the image repository is a server.

11. A method according to Claim 10,
wherein the server is a cable head end.

12. A method according to Claim 11,
5 wherein the upload component executes on a set top
box.

13. A method according to Claim 1, wherein
executing the upload component to retrieve the
10 digitally-captured image data and transmit the
retrieved image data to an image repository further
comprises:

retrieving the digitally-captured image
data into a buffer of a client computing device;
15 transmitting the digitally-captured image
data from the client computing device to the image
repository.

14. A method according to Claim 13,
20 wherein a server controls the timing of transmitting
the digitally-captured image data from the client
computing device to the image repository.

15. A method according to Claim 1, further
25 comprising:

displaying a user interface containing an
option to retrieve the digitally-captured image data
from storage accessible by the digital image storage
device;

responsive to a selection of the option,
retrieving all of the digitally-captured image data
from the accessible storage;

5 storing the retrieved image data in an
image repository; and

assigning a unique identifier to the
stored image data; and

storing the unique identifier in the
accessible storage.

10

16. A client computing device configured
to receive digitally-captured image data comprising:

15 a driver configured to receive type
information identifying a digital image storage
device;

a first portion of an upload component
configured to cause a second portion of the upload
component to be loaded in response to the type
information identifying the digital image storage
20 device;

20

the second portion of the upload component
configured to transmit digitally-captured image data
and associated information to an external storage
device.

25

17. A client computing device according to
Claim 16, wherein the external storage device is a
server, the second portion of the upload component
configured to transmit the digitally-captured image
30 data to the server via an upload proxy resident on
the server.

30

18. A client computing device according to Claim 16, further comprising a user interface for displaying an option to retrieve digitally-captured image data, in response to which the second portion of the upload component is configured to upload the digitally-captured image data to the external storage device and cause a unique identifier associated with the stored image data to be stored on storage media readable by the digital image storage device.

19. A method of uploading digitally-captured image data from storage of a digital image storage device, the method comprising:

displaying a user interface containing an option to retrieve digitally-captured image data accessible by a digital image storage device;

responsive to a selection of the option, retrieving all of the digitally-captured image data from storage of the digital image storage device;

storing the retrieved image data in an image repository; and

assigning a unique identifier to the stored image data; and

storing the unique identifier in storage of the digital image storage device.

20. A method according to Claim 19, wherein the image repository is a server.

21. A method according to Claim 19,
further comprising:

retrieving the stored digitally-captured
image data from the image repository using the
5 unique identifier in storage of the digital image
storage device.

22. A server configured to store
digitally-captured image data accessible by a
10 digital image storage device comprising:

a receiving component configured to
receive signals containing the digitally-captured
image data and associated information;

a token generator for generating a token
15 for the digitally-captured image data;

storage for storing the digitally-captured
image data and the token; and

a sending component configured to forward
the token for storage by the digital image storage
20 device.

23. A server according to Claim 22,
wherein the receiving component receives a request
containing the token, the server further configured
25 to retrieve the associated digitally-captured image
data from the server's storage and forward the image
data in response to the request.

24. A server according to Claim 22,
30 wherein the server is a cable head end.

25. A server according to Claim 23,
wherein the signals are received from a set top box.

26. A server according to Claim 25,
5 wherein the server is further configured to download
an upload component for execution on the set top
box.

27. A computer-readable memory medium in
10 which computer-executable process steps are stored,
the process steps for retrieving digitally-captured
image data from storage, wherein the process steps
comprise:

a detecting step to detect a signal from a
15 digital image storage device, the digital image
storage device capable of accessing the
digitally-captured image data;

an identifying step to identify a
component configured to upload digitally-captured
20 image data, the identification based on type
information received from the digital image storage
device;

an executing step to execute the upload
component to cause the digitally-captured image data
25 to be retrieved from the digital image storage
device and transmitted to an image repository.

28. A computer-readable memory medium
according to Claim 27, wherein the identifying step
30 to identify a component configured to upload the
digitally-captured image data is initiated in

response to detection of the digital image storage device and receipt of the type information from the digital image storage device.

5 29. A computer-readable memory medium according to Claim 27, wherein the digital image storage device is a digital camera.

10 30. A computer-readable memory medium according to Claim 27, wherein the digital image storage device is a scanning device.

15 31. A computer-readable memory medium according to Claim 27, wherein the digital image storage device is a video device.

20 32. A computer-readable memory medium according to Claim 27, wherein the digital image storage device is a removable storage media reader.

25 33. A computer-readable memory medium according to Claim 27, wherein the identifying step to identify a component for use in uploading digitally-captured image data further comprises:

an identifying step to identify the component using a mapping between the upload component and the type information received from the digital image storage device;

30 an identifying step to identify a location of the upload component; and

a retrieving step to retrieve the upload component from the identified location.

5 34. A computer-readable memory medium according to Claim 27, wherein the type identifies a manufacturer of the digital image storage device.

10 35. A computer-readable memory medium according to Claim 27, wherein the type identifies a model of the digital image storage device.

15 36. A computer-readable memory medium according to Claim 27, wherein the image repository is a server.

 37. A computer-readable memory medium according to Claim 36, wherein the server is a cable head end.

20 38. A computer-readable memory medium according to Claim 37, wherein the upload component executes on a set top box.

25 39. A computer-readable memory medium according to Claim 27, wherein the executing step to execute the upload component to retrieve the digitally-captured image data and transmit the retrieved image data to an image repository further comprises:

a retrieving step to retrieve the digitally-captured image data into a buffer of a client computing device;

5 a transmitting step to transmit the digitally-captured image data from the client computing device to the image repository.

40. A computer-readable memory medium according to Claim 39, wherein a server controls the
10 timing of transmitting the digitally-captured image data from the client computing device to the image repository.

41. A computer-readable memory medium
15 according to Claim 27, further comprising:

a displaying step to display a user interface containing an option to retrieve the digitally-captured image data accessible from storage by the digital image storage device;
20 responsive to a selection of the option, a retrieving step to retrieve all of the digitally-captured image data from accessible storage;

a storing step to store the retrieved
25 image data in an image repository;

an assigning step to assign a unique identifier to the stored digitally-captured image data; and

a storing step to store the unique
30 identifier in the accessible storage.

42. Computer-executable program code stored in a computer-readable medium, the computer-executable program code for retrieving digitally-captured image data from storage, wherein
5 the process steps comprise:

code to detect a signal from a digital image storage device, the digital image storage device capable of accessing the digitally-captured image data;
10 code to identify a component configured to upload digitally-captured image data, the identification based on type information received from the digital image storage device;
code to execute the upload component to
15 cause the digitally-captured image data to be retrieved from the digital image storage device and transmitted to an image repository.

43. Computer-executable program code
20 according to Claim 42, wherein the code to identify a component configured to upload the digitally-captured image data is initiated in response to detection of the digital image storage device and receipt of the type information from the
25 digital image storage device.

44. Computer-executable program code according to Claim 42, wherein the digital image storage device is a digital camera.

45. Computer-executable program code according to Claim 42, wherein the digital image storage device is a scanning device.

5 46. Computer-executable program code according to Claim 42, wherein the digital image storage device is a video device.

10 47. Computer-executable program code according to Claim 42, wherein the digital image storage device is a removable storage media reader.

15 48. Computer-executable program code according to Claim 42, wherein the code to identify a component for use in uploading digitally-captured image data further comprises:

20 code to identify the component using a mapping between the upload component and the type information received from the digital image storage device;

 code to identify a location of the upload component; and

 code to retrieve the upload component from the identified location.

25

49. Computer-executable program code according to Claim 42, wherein the type identifies a manufacturer of the digital image storage device.

50. Computer-executable program code according to Claim 42, wherein the type identifies a model of the digital image storage device.

5 51. Computer-executable program code according to Claim 42, wherein the image repository is a server.

 52. Computer-executable program code
10 according to Claim 51, wherein the server is a cable head end.

 53. A computer-readable memory medium according to Claim 52, wherein the upload component
15 executes on a set top box.

 54. Computer-executable program code according to Claim 42, wherein code to execute the upload component to retrieve the digitally-captured
20 image data and transmit the retrieved image data to an image repository further comprises:

 code to retrieve the digitally-captured image data into a buffer of a client computing device;
25 code to transmit the digitally-captured image data from the client computing device to the image repository.

 55. Computer-executable program code
30 according to Claim 54, wherein a server controls the timing of transmitting the digitally-captured image

data from the client computing device to the image repository.

56. Computer-executable program code
5 according to Claim 42, further comprising:
code to display a user interface
containing an option to retrieve the
digitally-captured image data from storage
accessible by the digital image storage device;
10 responsive to a selection of the option,
code to retrieve all of the digitally-captured image
data from the accessible storage;
code to store the retrieved image data in
an image repository;
15 code to assign a unique identifier to the
stored image data; and
code to store the unique identifier in the
accessible storage.

57. A computer-readable memory medium in
20 which computer-executable process steps are stored,
the process steps for uploading digitally-captured
image data from storage of a digital image storage
device, wherein the process steps comprise:
25 a displaying step to display a user
interface containing an option to retrieve
digitally-captured image data accessible by a
digital image storage device;
responsive to a selection of the option, a
30 retrieving step to retrieve all of the

digitally-captured image data from storage of the digital image storage device;

a first storing step to store the retrieved image data in an image repository; and

5 an assigning step to assign a unique identifier to the stored image data; and

a second storing step to store the unique identifier in storage of the digital image storage device.

10

58. A computer-readable memory medium according to Claim 57, wherein the image repository is a server.

15

59. A computer-readable memory medium according to Claim 57, further comprising:

a retrieving step to retrieve the stored image data from the image repository using the unique identifier in storage of the digital image storage device.

20

60. Computer-executable program code stored on a computer-readable medium, said computer-executable code for uploading

25 digitally-captured image data from storage of a digital image storage device, wherein the computer-executable program code comprise:

code to display a user interface containing an option to retrieve digitally-captured

30 image data accessible by the digital image storage device;

responsive to a selection of the option,
code to retrieve all of the digitally-captured image
data from storage of the digital image storage
device;

5 code to store the retrieved image data in
an image repository; and

 code to assign a unique identifier to the
stored image data; and

 code to store the unique identifier in
10 storage of the digital image storage device.

61. Computer-executable program code
according to Claim 60, wherein the image repository
is a server.

15

62. Computer-executable program code
according to Claim 60, further comprising:

 code to retrieve the stored image data
from the image repository using the unique
20 identifier in storage of the digital image storage
device.

ABSTRACT

The present invention relates to the retrieval and storage of image data captured using a digital image acquisition device such as a still camera, a video camera or a scanning device, and of image data stored on removable storage media such as compact flash, smart and memory stick media. An image retrieval architecture is described such that image data may be retrieved from any digital image acquisition device and/or from removable storage media, and a "one-click" operation is available to minimize an amount of interaction needed to retrieve image data from the digital image acquisition device.

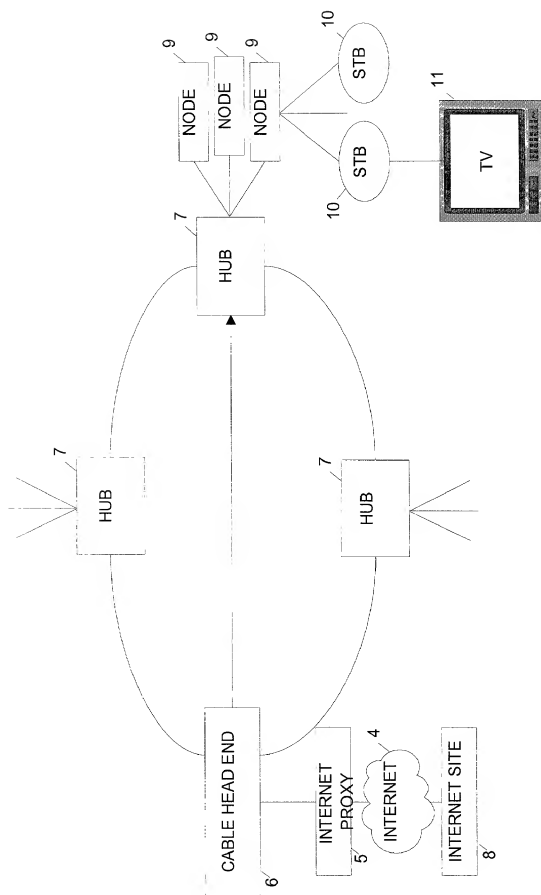


Fig. 1

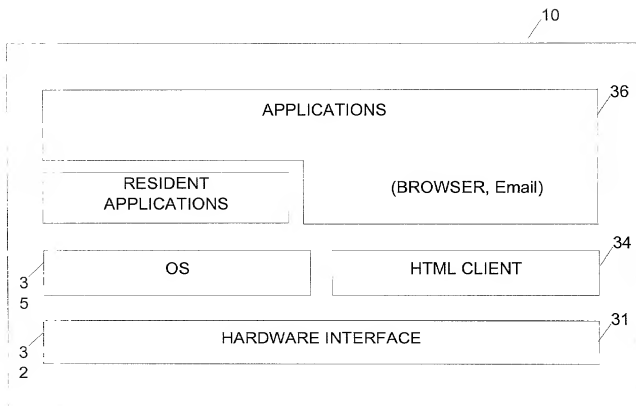


Fig. 2

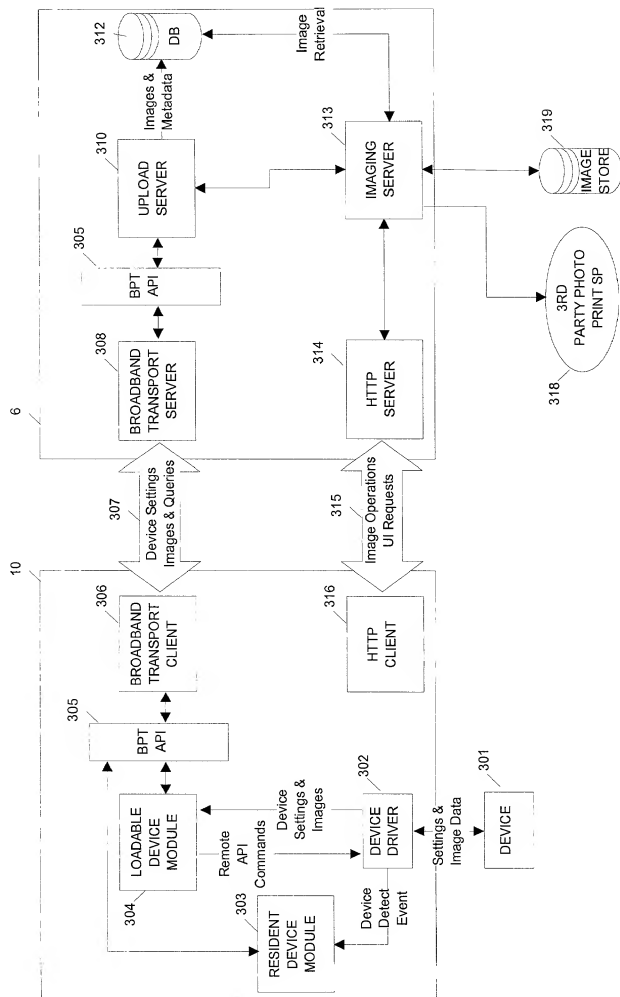


Fig. 3

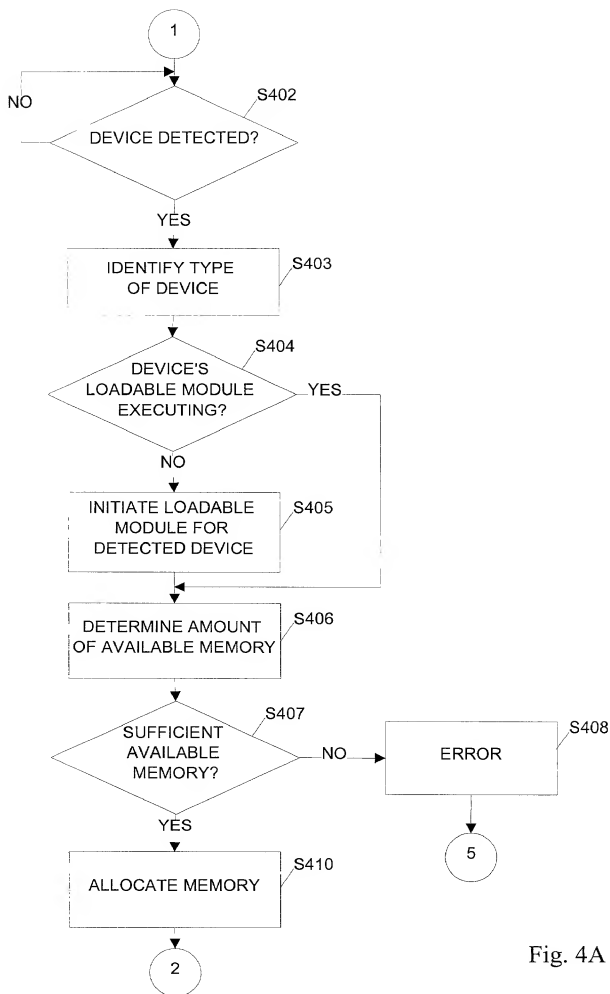
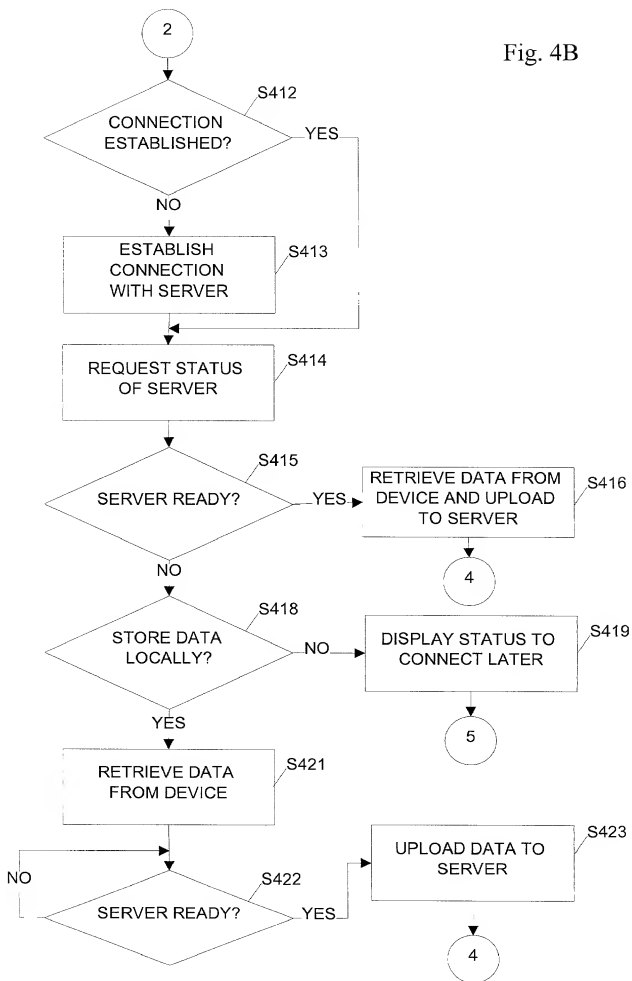


Fig. 4A

Fig. 4B



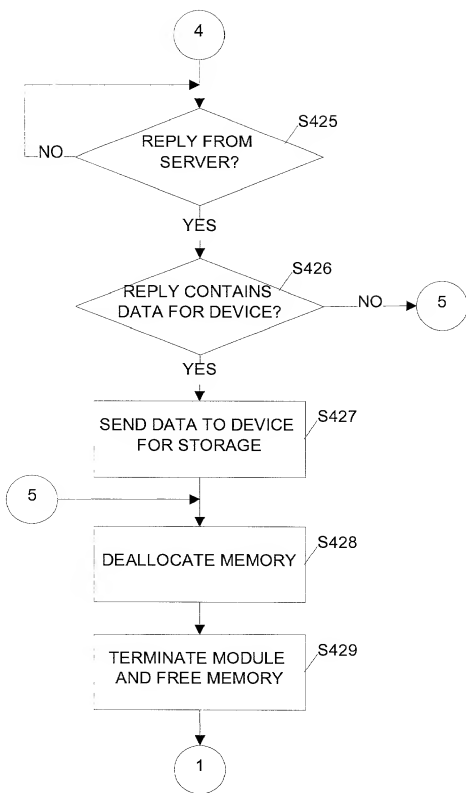


Fig. 4C

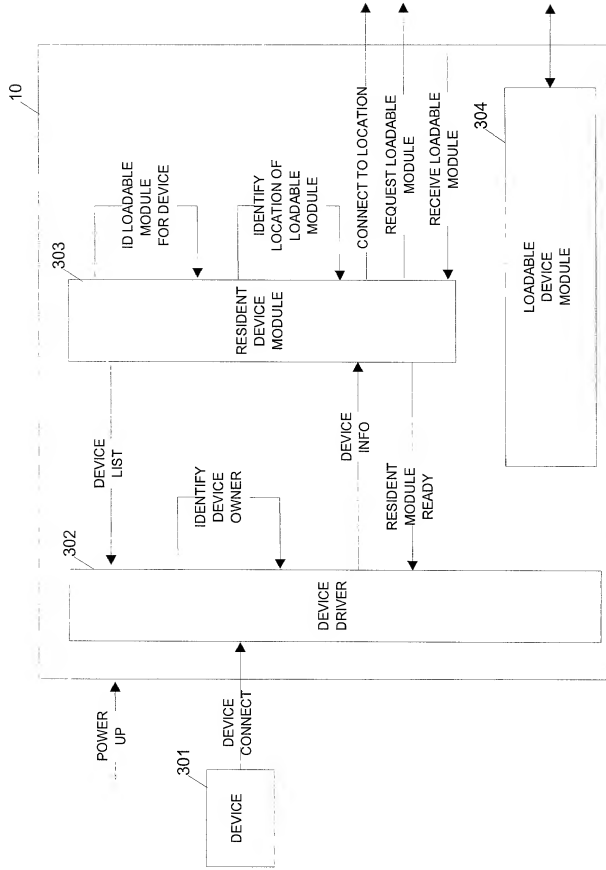


Fig. 5

601

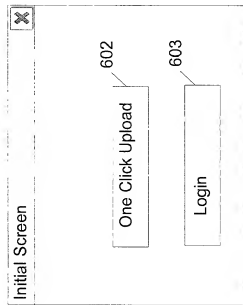


Fig. 6A

621

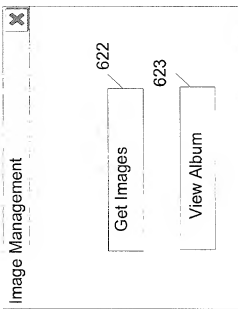


Fig. 6C

611

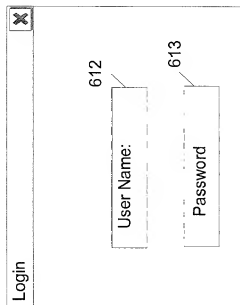


Fig. 6B

631

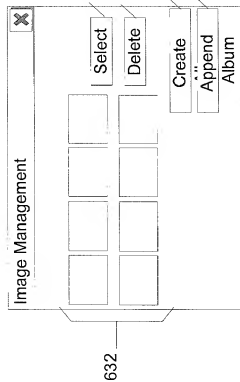


Fig. 6D

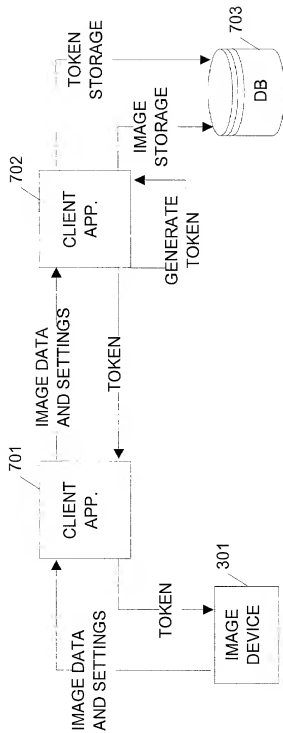


Fig. 7A

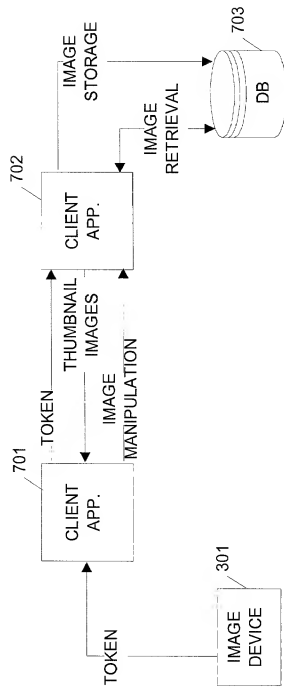


Fig. 7B